From Trace to Trust: Governing Copyright in Generative AI

Douglas C. Youvan

doug@youvan.com

www.youvan.ai

October 20, 2025

When a large language model ingests copyrighted works, their "traces" diffuse into weights, optimizer state, and behavior in ways that are powerful for learning but tricky for law and policy. Unlike a database, a model does not store verbatim pages so much as a statistical imprint—yet rare or repeated passages can still be memorized and reproduced. Output scanners help, but they police *behavior*, not *provenance*. At the same time, U.S. fair-use doctrine weighs purpose, amount, and market effects rather than simple thresholds, and recent decisions caution against broad claims of "transformative" use when outputs substitute for originals. This paper synthesizes a practical governance stack: data hygiene and deduplication; memorization auditing and canary tests; differential privacy and targeted unlearning; post-decode similarity scanning with attribution-first pathways; and decision logging that creates defensible proofs of diligence. We map each control to the four fair-use factors and to concrete product choices (decoding constraints, regeneration-on-hit policies, and user-visible citation modes). Our thesis is simple: there is no single moment when traces vanish, but there is a reproducible way to lower legal, ethical, and reputational risk—moving organizations from "hope it's fine" to *operational fairness* backed by tests, thresholds, and documented intent.

Keywords: fair use, generative AI, copyrighted material, memorization, differential privacy, machine unlearning, output scanning, similarity detection, attribution, governance, audits, canary exposure, membership inference, policy, market substitution, data hygiene, Suchir Balaji.

A collaboration with GPT-5. 56 pages. CC4.0.

Outline

1. Problem Statement and Scope
   1.1 What “traces” mean technically vs. legally
   1.2 Training vs. generation: two different risk surfaces
   1.3 Stakeholders: users, rights-holders, labs, regulators
2. Conceptual Foundations
   2.1 Statistical imprint, memorization, and rare-string effects
   2.2 Model artifacts: checkpoints, optimizer state, eval logs
   2.3 Idea–expression boundary and why style ≠ protected expression
3. Legal Frame: Fair Use in Practice (U.S.-centric, globally aware)
   3.1 The four factors and their modern reading
   3.2 Search/indexing precedents vs. substitutive outputs
   3.3 Lessons from recent jurisprudence for generative systems
   3.4 Risk taxonomy by domain (lyrics/poetry, code, news, images)
4. Data Governance Before Training
   4.1 Source acquisition, licensing tiers, and deny/allow lists
   4.2 Deduplication at scale (exact, fuzzy, semantic)
   4.3 Sensitive-genre throttles and sampling strategies
   4.4 Provenance logging and reproducible data slices
5. Training-Time Controls
   5.1 Memorization audits: canaries, exposure metrics, stress tests
   5.2 Differential privacy (DP-SGD): guarantees, costs, and tuning
   5.3 Targeted and approximate machine unlearning: capabilities and limits
   5.4 Curriculum and objective tweaks that reduce overfitting
6. Generation-Time Controls (Behavioral Guardrails)
   6.1 Decoding constraints: repetition penalties, outline→paraphrase flows
   6.2 Post-decode similarity scanning: n-gram, fuzzy hash, embedding search
   6.3 Policy actions: block, regenerate, or quote + attribute
   6.4 User modes: “Attribute Mode” vs. “Rewrite Mode”
7. Attribution and Transparency
   7.1 When short quotation with citation is preferable to suppression
   7.2 UX for inline citations and source cards
   7.3 Respecting paywalls and market channels

8. Proof of Diligence: Audits and Logs
   8.1 Decision logs: hashes, thresholds, and sources checked
   8.2 Membership-inference testing and red-team drills
   8.3 Periodic memorization re-audits and model lifecycle checkpoints

9. Operational Playbooks
   9.1 “Tainted checkpoint” response: rollback, scrub, retrain, re-audit
   9.2 Incident handling for reported overlaps (triage → remedy → report)
   9.3 Vendor and model-hub requirements (SLAs, attestations, test suites)

10. Policy Alignment and Market Effects
   10.1 Mapping controls to the four fair-use factors
   10.2 Measuring substitution risk and offering creator pathways
   10.3 Interop with licensing, collective management, and opt-out/opt-in

11. Case Studies (Patterns, Not Parties)
   11.1 News paragraph overlap: regenerate vs. cite
   11.2 Song lyrics request: denylist + educational excerpting
   11.3 Code snippet similarity: function-level scan and rewrite

12. Limits, Open Problems, and Research Agenda
   12.1 Formal unlearning at frontier scale
   12.2 High-utility DP for multimodal models
   12.3 Standardized exposure benchmarks and public audits

13. Conclusion: From Trace Anxiety to Trusted Practice
   13.1 No “vanish point,” but a defensible process
   13.2 Shipping *operational fairness* as a product feature
   13.3 A roadmap for verifiable, creator-respecting AI

Epilogue: The Suicide of OpenAI’s Suchir Balaji

Appendices (for later expansion)
A. Memorization audit checklist
B. Output-scanner reference thresholds by domain
C. Model-release attestation template (data, audits, controls)

References

1. Problem Statement and Scope

1.1 What “traces” mean technically vs. legally

Technically, a “trace” is any residual signal of a training example that persists in a model’s artifacts or behavior. Traces arise at multiple layers: (a) storage artifacts (raw/preprocessed files, shards, caches, backups); (b) model internals (weights, optimizer state, learned embeddings), where examples contribute infinitesimal but sometimes *memorized* influence; and (c) observable behavior (verbatim or near-verbatim reproduction, distinctive structure, or characteristic error patterns). A trace can be quantitative (exposure scores, membership inference risk) or qualitative (passage regurgitation, idiosyncratic phrasing).

Legally, “trace” is not a doctrine. Courts assess *uses* of copyrighted expression through the fair-use factors and related doctrines (substantial similarity, market harm, etc.). The law does not police the mere *existence* of a statistical imprint; it examines *copying* and *use*—how the material was obtained, what portion was taken, how it is disclosed, and whether the output substitutes for the original. Thus, a technical trace is not automatically a legal problem; conversely, a legally problematic output may occur even when technical “trace” metrics are low. The core tension is that engineers reason in gradients and probabilities, whereas law reasons in purposes, effects, and evidence.

1.2 Training vs. generation: two different risk surfaces

Training-time risk centers on *provenance and process.* Key questions: Was the data lawfully acquired or licensed? Were sensitive genres controlled (lyrics, code, paywalled text)? Were deduplication and anti-memorization steps taken? Do logs show purpose-limited, non-expressive analysis (e.g., learning features rather than publishing copies)? Remedies at this layer are architectural: dataset curation, dedup, canaries, differential privacy, and—if necessary—unlearning or retraining from a pre-taint checkpoint.

Generation-time risk centers on *behavior and impact.* The issue is whether outputs reproduce protected expression or function as substitutes. Controls are behavioral: decoding constraints (reduce long n-gram reuse), similarity scanning (exact/fuzzy/semantic), and policy actions (block, regenerate, or quote with citation). Importantly, output controls cannot “clean” training provenance; they only govern what leaves the system.

Interaction effects matter. A model trained with robust anti-memorization typically reduces downstream overlap incidents, lowering enforcement and reputational risk. Conversely, even a carefully sourced dataset can yield risky outputs if decoding and scanning are lax. Effective

governance treats the two surfaces as complementary: *prevent at train time, police at generation time, prove diligence across both.*

1.3 Stakeholders: users, rights-holders, labs, regulators

- Users (individuals & enterprises): Seek useful, fluent outputs with minimal legal exposure. They need clear UX affordances (quote-and-cite vs. rewrite), transparent flags when similarity is detected, and predictable remediation (regenerate-on-hit). Enterprises also need audit trails to defend downstream use.
- Rights-holders (authors, publishers, labels, studios, software vendors): Care about substitution, attribution, and licensing pathways. They will push for either participation (licensing/attribution revenue) or protection (opt-outs, denylists, strong scanning). Trust increases when models avoid long verbatim passages and offer creator-respecting modes (attribution-first for short excerpts).
- Labs and model vendors: Own the training provenance, controls, and attestations. Their priorities are performance, compliance, and defensibility: clean data pipelines, exposure/membership audits, DP or equivalent safeguards where feasible, and well-documented incident response for reported overlaps.
- Regulators and courts: Focus on process, purpose, and market effects. They will ask whether developers applied proportionate safeguards (dataset hygiene, dedup, audits, deletion of raw corpora), whether outputs substitute for protected works, and whether there are accessible redress mechanisms for rights-holders.
- Bridges between stakeholders: Standards bodies, collective management organizations, and marketplaces can convert conflict into interop—e.g., registries for opt-in/opt-out, standardized exposure benchmarks, and evidence formats (hashes, logs) that make diligence verifiable.

Scope boundary. This paper addresses *text-centric* generative systems and adjacent modalities where the same principles apply (e.g., code, some image/text pairs). It does not attempt to resolve moral rights, database rights in the EU, or non-copyright regimes (e.g., trade secrets) except where they intersect operational controls. The goal is pragmatic: define “trace” in actionable technical terms, map it to legal risk at both training and generation layers, and specify controls and proofs that stakeholders can adopt today.

2. Conceptual Foundations

2.1 Statistical imprint, memorization, and rare-string effects

Statistical imprint. Gradient descent does not store documents; it adjusts parameters to reduce loss on next-token prediction. Every example contributes a tiny update, so most influence is *distributed*—patterns become capabilities (syntax, facts, discourse moves). This diffuse signal is the model's statistical imprint: information present only as useful directions in weight space.

Memorization. Despite diffusion, models can memorize *particular* sequences. Empirically, memorization risk rises with: (a) duplication of passages in the corpus, (b) rarity of the sequence (low natural probability), (c) overfitting from long training on narrow domains, and (d) high capacity relative to data diversity. Memorized spans are typically long contiguous n-grams (e.g., 50–200+ tokens) that can be elicited by targeted prompts.

Rare-string effects. Highly unique strings (poetry, lyrics, legal boilerplate, API keys, puzzles, proprietary code headers) behave like "keys." When they appear repeatedly—or even once but with strong local gradients—they can be extracted with cueing ("continue this stanza…") or by brute-forcing prefix prompts. This motivates deduplication (exact/fuzzy/semantic), exposure testing (canaries with known counts), and decoding constraints that penalize long n-gram reuse.

Operational implications.

- Treat duplication rate and uniqueness scores as first-class dataset metrics.
- Run memorization audits: insert canaries at known frequencies, then measure exposure under diverse prompts.
- Favor curriculum/regularization that reduces overfit (mixing, dropout, noise, early-stop checkpoints).
- Use post-decode scanners to catch residual long-span overlap before output leaves the system.

2.2 Model artifacts: checkpoints, optimizer state, eval logs

Checkpoints. A checkpoint captures weights (and sometimes embeddings, tokenizer state). It is the durable carrier of the statistical imprint and any memorized spans. If a dataset segment is later deemed out-of-scope, the safest remedy is to retrain from a *pre-taint* checkpoint or apply targeted unlearning (still approximate at LLM scale).

Optimizer state. Modern training (e.g., Adam/Adafactor) maintains first/second-moment accumulators. These states encode a history of gradients—in effect, a memory of which

directions were repeatedly reinforced. Even if you freeze weights, lingering optimizer moments can re-introduce bias when training resumes. For strong provenance hygiene, snapshot and govern both weights and optimizer state; if you roll back, roll both.

Eval and telemetry artifacts. Validation dumps, qualitative eval notebooks, prompt/response logs, and red-team transcripts can contain verbatim snippets from the training or evaluation corpora. Because these artifacts are human-readable, they are a frequent *leak path*. Apply the same controls you apply to datasets: access control, retention limits, hashing, and PII/IP screening.

Operational implications.

- Define “artifact of record”: weights + optimizer moments + tokenizer/versioning + data slice hashes.
- Maintain lineage manifests tying each checkpoint to dataset commits and audit results.
- Treat eval logs as sensitive content; minimize verbatim copies, scrub on export, and time-limit retention.
- When scrubbing provenance, verify that backups and mirrors (object stores, scratch disks, CI caches) are covered.

2.3 Idea–expression boundary and why style ≠ protected expression

The boundary. Copyright protects expressive form, not ideas, facts, systems, or methods. In practice, two analyses matter:

- Protected expression: specific wording, structure, sequence, and arrangement that reflect authorial choices.
- Unprotected substance: the underlying ideas, facts, procedures, and general themes.

Style vs. expression. “Write in the style of X” typically targets features (tone, cadence, vocabulary distribution, rhetorical moves) rather than reproducing specific expression (the sentences X actually wrote). Style—as a set of statistical properties—is concept-like, not a verbatim copy. Risk arises when a style prompt plus the model’s memorization yields long, close paraphrases or contiguous overlap with a known work.

Practical tests for outputs.

- Similarity span & density: Long contiguous overlaps (e.g., 40+ tokens) in multiple places suggest protected expression, not mere style.

- Substitutive effect: If the output could replace the original for the usual audience (lyrics, news ledes, code blocks), risk is high even if wording shifts.
- Boilerplate filter: Common phrases, definitions, and clichés trigger false positives; maintain a boilerplate whitelist to avoid over-blocking.

Operational implications.

- Default to quote-and-cite for short, high-value excerpts (criticism, scholarship, research) rather than suppressing everything similar.
- For risky domains (lyrics/poetry/scripts/code), adopt stricter thresholds, finer-grained scanning, and regenerate-on-hit policies.
- Communicate the boundary in UX: offer "Attribute Mode" (allow short excerpts with citation) vs. "Rewrite Mode" (force paraphrase and lower overlap), so users choose behavior aligned with their use case.

Summary. The core technical picture is: most training influence is diffuse (statistical imprint), but memorization of rare or repeated spans creates extractable risk; that risk lives across artifacts (weights, moments, logs). The legal picture hinges on expression vs. ideas and market substitution. Governance links them: shape datasets to minimize memorization, control artifacts to limit leakage, and shape outputs to respect the idea–expression line.

3. Legal Frame: Fair Use in Practice (U.S.-centric, globally aware)

3.1 The four factors and their modern reading

U.S. fair use is a case-by-case balancing of four statutory factors: (1) purpose and character of the use (including commerciality and transformation), (2) nature of the work, (3) amount/substantiality used, and (4) market effect. Courts weigh them together rather than as bright-line rules. For AI, factor (1) turns on whether copying is for non-expressive analysis (e.g., feature learning/search) versus expressive substitution; factor (3) is sensitive to wholesale ingestion; and factor (4) scrutinizes substitutive outputs. Recent Supreme Court guidance narrows expansive "transformative" claims when the secondary use shares the original's commercial purpose (e.g., licensing an image to illustrate a magazine story).

3.2 Search/indexing precedents vs. substitutive outputs

Courts have long treated intermediate copying for search or analysis as favoring fair use when public disclosure is limited and the function is new:

- Google Books: scanning entire books to enable full-text search and snippet view was fair use—transformative indexing with limited output; no evidence of market substitution.
- Image search thumbnails: using reduced-size images to power discovery was fair use because the search function differed from consuming the full images.
- Turnitin plagiarism detection: archiving student papers for non-expressive comparison was fair use.

By contrast, when the secondary use competes in the same expressive market, fair use weakens. Warhol v. Goldsmith held that licensing a Warhol image derived from a photograph to illustrate a magazine story shared the original's purpose, weighing against fair use. For AI, this cautions against outputs that stand in for news articles, lyrics, photos, or code sold/licensed in the same channels.

3.3 Lessons from recent jurisprudence for generative systems

- Training-time claims are still developing. U.S. cases against model developers (e.g., NYT v. OpenAI/Microsoft, artist suits against image models) remain in pretrial or mixed procedural stages. Judges have allowed some copyright theories to proceed (e.g., direct infringement) while trimming others; no definitive nationwide ruling has blessed or banned LLM training as fair use. Expect fact-intensive inquiries into purpose, amount, and market effects.
- UK proceedings (Getty v. Stability) show a different track: case-management orders and interlocutory decisions are moving the litigation toward trial, highlighting jurisdictional and database right nuances outside U.S. law.
- Early U.S. rulings in artist/news cases generally permit core copyright claims to advance (e.g., after motions to dismiss), signaling courts want a fuller factual record on datasets, memorization, and substitution before deciding fair use.
- Warhol's shadow: Factor-one analysis now emphasizes purpose overlap and commercial licensing. For model outputs, even if style differs, substitutive licensing (e.g., magazine covers, stock photos) cuts against fair use.

3.4 Risk taxonomy by domain (lyrics/poetry, code, news, images)

- Lyrics/Poetry/Scripts (highest risk): Short works where small spans can be "substantially similar." Markets are license-driven, and verbatim or close paraphrase often substitutes for the original. Strong denylists, aggressive overlap thresholds, and *quote-with-citation for criticism/education only* are prudent. Warhol-style purpose overlap is common when users want "the song/the poem itself."
- News/Nonfiction: Facts aren't protected, but expression and selection are. Risk spikes with lede-paragraph reproduction and paywalled content. Courts in ongoing news cases are examining whether outputs replace the need to access the article. Use short quotations with citation for commentary; avoid long contiguous overlap.
- Software/Code: Function matters, but creative structure/selection/arrangement of non-trivial code can be protected. Exact/near-exact reproduction of functions or headers is risky and substitutive; scanners should operate at function-block granularity. (Some suits against code copilots are pending; doctrines from search cases don't automatically carry over.)
- Images/Thumbnails/Derivatives: Search thumbnails can be fair use (transformative indexing), but full-fidelity or market-ready outputs that track a specific source are riskier, especially where stock-image licensing is the market. Pending Getty v. Stability proceedings underscore this risk.

Takeaway. For AI, fair use is strongest where the system's use is analytical and non-expressive (training as indexing/feature learning with controlled disclosure), and weakest where outputs compete with or reproduce protected expression in the same market. Align technical controls (deduplication, memorization audits, output scanning, citation UX) with these legal gradients to reduce exposure and demonstrate diligence.

4. Data Governance Before Training

4.1 Source acquisition, licensing tiers, and deny/allow lists

Acquisition policy (first principles).

- Prefer contracted/consented sources (owned, licensed, contributed) over scraped/ambiguous.
- Record, per item: source_id, URL/URI, license class, rights-holder, acquisition path, timestamp, jurisdiction, and redistribution constraints.

Licensing tiers (example policy ladder).

- Tier A — Owned/Assigned: company-owned docs, employee contributions, open hardware/software you own. *Full use; redistributable embeddings/checkpoints per policy.*
- Tier B — Contract/Licensed: partner datasets, content marketplaces, CMO bundles. *Use per license; include contract_id, term, revocation clause.*
- Tier C — Truly Open: PD/CC0, permissive licenses (CC-BY with attribution pipeline, OSI licenses for code). *Require attribution hooks if needed.*
- Tier D — Ambiguous/Disputed: unclear rights, paywalled-to-public leaks, scraped with robots exclusion. *Quarantine; do not train.*
- Tier E — Prohibited: explicit opt-outs, sensitive personal data, DRM-protected leaks, known litigious catalogs (e.g., lyrics). *Denylist.*

Deny/allow mechanics.

- Denylist by *work*, *catalog*, *domain*, *pattern* (e.g., /lyrics/, /fulltext_pdf/), *publisher*, *jurisdiction*.
- Allowlist of trusted domains, registries, or verified contributors (with attested provenance).
- Fail-closed ingestion: anything without a resolvable license class defaults to blocked.

Revocation & takedown.

- Maintain revocation manifests (hash sets + pointers). If a license is pulled, you can re-slice data (see 4.4) and roll back to a *pre-taint* checkpoint.

4.2 Deduplication at scale (exact, fuzzy, semantic)

Why it matters. Dupes inflate memorization risk, skew sampling, and waste compute.

Three-layer dedup stack.

1. Exact
    - Unit: normalized text *document/chunk* (e.g., 2–4k tokens).
    - Technique: cryptographic content hash (SHA-256/XXH3) after normalization (lowercase, whitespace, unicode NFC, boilerplate strip).

    - Action: drop duplicates; keep a provenance fan-in list for the survivor.
2. Fuzzy / Near-duplicate
    - Unit: sliding windows (e.g., 256–512 tokens, stride 64–128).
    - Technique: shingling + MinHash/LSH or SimHash; threshold Jaccard ≥ 0.8 (text), ≥ 0.9 (code, legal).
    - Action: cluster; keep one representative per cluster; retain edges in a dup graph for audits.
3. Semantic
    - Unit: chunk-level embeddings (same windows).
    - Technique: ANN index (FAISS/ScaNN); cosine ≥ 0.92 for narrative, ≥ 0.97 for boilerplate/code.
    - Action: down-weight or drop highly similar items across sources; mark as semantic-near-dup (SND).

Corner cases & hygiene.

- Boilerplate (licenses, TOCs, navigation chrome): maintain a *boilerplate whitelist* to avoid over-flagging and suppress its training weight to near-zero.
- Canonicalization: normalize quotes, hyphens, numerals; strip tracking strings; detect optical artifacts (PDF-to-text).
- Metrics to track: % exact dupes removed, % near-dup clusters, rare-string exposure (e.g., proportion of 20-gram once-seen vs. many-seen).
- Artifacts: persist dup clusters and keep lists so you can reproduce the survivor set deterministically.

4.3 Sensitive-genre throttles and sampling strategies

High-risk genres (typical): lyrics/poetry/scripts, news paywalls, code under restrictive licenses, proprietary manuals, academic PDFs with non-open licenses, PII-bearing corpora.

Genre throttles (policy levers).

- Quota caps: e.g., lyrics ≤ 0.01% tokens; poetry/scripts ≤ 0.1%; paywalled news ≤ 0.5% with license.

- Down-weighting: scale loss contribution for risk-heavy genres by $\alpha \in [0.1, 0.5]$.
- Contiguity limits: clip max contiguous tokens from a single risky document per batch (e.g., ≤ 256).
- Context mixing: interleave risky windows with diverse, licensed text to reduce overfit.

Sampling strategies (practical).

- Stratified sampling by license tier + domain, targeting a fixed token budget per stratum.
- Hash-based deterministic sampling (see 4.4) so the same rules pick the same items across rebuilds.
- Curriculum: front-load broad, low-risk corpora; introduce higher-risk/complex registers later with reduced weight.
- Diversity guards: enforce source and domain entropy in each epoch (e.g., max 1% tokens from any single domain in a given 10M-token window).
- Rare-string brakes: detect unique/low-frequency 20–50-grams; cap their repetition across epochs to prevent memorization keys.

Audit hooks.

- Per-genre exposure tests (canaries similar to real items) with target exposure < threshold (e.g., < 10% for top-k).
- Per-genre overlap spot checks with your post-decode scanner before opening training to the next phase.

4.4 Provenance logging and reproducible data slices

Reproducibility goals. Anyone with access to the store should be able to rebuild the *exact* training slice for a checkpoint using a manifest + code + seed.

Data model (per item).

- content_hash (normalized), raw_hash (original), size, mime, language
- source_uri/domain, acquisition_ts, crawler_version (if any)
- license_class (A–E), contract_id/license_url, jurisdiction
- dup_cluster_id (exact/near/semantic), boilerplate_flag, pii_flag

- risk_genre (enum), sampling_weight, throttle_class
- processing_commits (preprocess v#, dedup v#, boilerplate v#)

Slice manifests.

- A deterministic query over the catalog (SQL or DSL) that encodes:
    - license filters (Tier ∈ {A,B,C}, exclude D/E),
    - time bounds (e.g., acquisition_ts < 2025-09-01),
    - dedup policy versions, genre throttles, language mix, max/domain caps,
    - PRNG seed and hash-based sampler (e.g., keep_if hash(item_id || seed) % 10000 < k).
- Output a manifest file: ordered list of (item_id, window_offsets), with total token count, domain distribution, and risk summary.

Lineage & attestations.

- For each checkpoint: store the slice_manifest_hash, code commit SHAs, tokenizer version, hyperparams, and optimizer state hash.
- Emit a provenance attestation (machine-readable) that can be shared with partners/regulators without exposing raw content (Bloom or Merkle proofs of inclusion/exclusion by domain/license class).

Revocation & rebuild.

- Maintain revocation lists (sets of content_hash/raw_hash). Rebuild the slice by:
    1. applying the same manifest query but subtracting revocations,
    2. producing a delta report (removed counts by domain/genre),
    3. rolling back to the last pre-taint checkpoint (weights + optimizer), or applying validated unlearning,
    4. re-running memorization and overlap audits.

Security & retention.

- Encrypt catalogs and manifests at rest; restrict PII/secret-bearing items to separate enclaves.

- Time-bound retention for raw/preprocessed data; keep only hashes + manifests long-term where feasible.
- Log all reads/writes to the catalog; treat eval/export notebooks as sensitive (short TTL, scrubbed on export).

Operational checkpoint (ready-to-train gate).

- ✅ All items have license_class and pass denylist checks.
- ✅ Dedup metrics within targets; rare-string exposure below threshold.
- ✅ Genre throttles enforced; sampling report meets diversity caps.
- ✅ Slice manifest sealed (hash), lineage recorded, revocation path tested.
- ✅ Baseline memorization audits pass before first gradient step.

5. Training-Time Controls

5.1 Memorization audits: canaries, exposure metrics, stress tests

Goal. Detect and bound unintended verbatim recall *before* the model ships.

Canaries (plant, then probe).

- Insertion. Add K synthetic passages (200–400 tokens each) that look like high-risk genres (lyrics, poetry, legalese, code headers), each repeated r times. Example: K=1,000; r in {1, 2, 5, 10}.
- Seeding. Generate unique 20–50-gram substrings per canary; keep a registry with content_hash and counts.
- Placement. Disperse across shards and epochs; never co-locate many canaries in one batch.

Exposure metric (copy-safe).

- For each canary c with count r, compute exposure E(c) := log_2(|U|) − log_2(rank_c), where |U| is the candidate space for sequences of the same length and rank_c is the rank of the true continuation under greedy or beam search. Target: E(c) below a fixed bound for all r; exposure should grow sublinearly with r.

- Pass criteria. No canary yields > 50 contiguous tokens under any of: greedy, nucleus, or beam search when primed with its first 30–60 tokens.

Stress tests (automated).

- Prefix attack. Prompt with the first N tokens of each canary; sweep N in {10, 20, 40, 60}.
- Semantic cueing. Use embedding nearest-neighbors to craft paraphrased cues (e.g., “Continue the following poem about X in Y’s meter…”).
- Brute force. Randomly sample prefixes from the training set (non-canaries); measure maximum contiguous overlap of completions with the corpus.
- High-T, low-penalty. Decode at T in {0.7, 1.0, 1.3} with minimal repetition penalties to probe worst-case recall.

Audit outputs.

- Metrics dashboard. Max span overlap, distribution of E(c), % prompts eliciting >= 50-token overlap, per-genre breakdown.
- Gate. Training may proceed only if all canary exposures and overlap rates are under thresholds; otherwise adjust data (dedup/throttles) or objectives and re-train.

5.2 Differential privacy (DP-SGD): guarantees, costs, and tuning

What DP gives. A formal upper bound on how much any single training example can change the distribution of model outputs. Typical statement: training is (epsilon, delta)-DP for some small delta (e.g., 1e-6). Lower epsilon ⇒ stronger privacy.

Mechanics (mini-batch DP-SGD).

- Clip per-example gradients to L2 norm C.
- Add Gaussian noise with std = sigma * C to the *sum* of clipped gradients.
- Account privacy loss over steps with a moments/Rényi accountant to compute epsilon at the chosen delta.
- Key knobs: clipping norm C, noise multiplier sigma, sampling rate q (batch_size / dataset_size), total steps T.

Tuning playbook.

- Start with C chosen from a short sweep that preserves perplexity on public dev sets; typical C in [0.1, 1.0] (normalized).
- Choose sigma to target epsilon in [3, 8] at delta ~ 1e-6 for a single training run; verify with the accountant.
- Use large batches and Poisson sampling to lower epsilon at fixed utility.
- Noise annealing (high early, taper late) can recover utility, but the privacy proof must cover the entire schedule.
- Apply gradient accumulation under DP carefully; only the *sampled* per-example grads get clipped/noised.

Costs.

- Utility hit. Expect higher perplexity and weaker long-range coherence at strong privacy.
- Compute. Per-example gradients and accounting add overhead; plan for 1.2x–2x training cost.
- Scope. DP protects against *membership inference*; it does not by itself prevent all forms of long-span overlap if the dataset is highly duplicated or uniquely phrased. You still need dedup and audits.

Ship criteria (suggested).

- Publish (epsilon, delta) with accountant details, C, sigma, q, T.
- Show *delta utility loss* vs. non-DP baseline on public benchmarks, plus your memorization audit deltas.

5.3 Targeted and approximate machine unlearning: capabilities and limits

Problem. Remove the influence of specific items after training (revocation requests, license pullbacks).

Families of methods.

- SISA-style partitioning. Shard data into disjoint slices; train ensembles or staged models so you can retrain only affected shards. Pros: simple accounting; Cons: partial coverage, infra complexity.

- Gradient negation / ascent. Fine-tune to push weights away from the gradients induced by the target set. Needs stored per-example gradients or surrogates; risks collateral damage.
- Regularization-based forgetting. Add penalties that preserve performance on a *keep* set while degrading fit on the *forget* set (e.g., Fisher/EWC-like constraints).
- Distillation with exclusion. Distill from the old model using a *reference corpus minus the forget set*, optionally adding anti-features to suppress recall.
- Retrain from pre-taint checkpoint. The most reliable, but expensive; requires strong lineage to identify the last clean state.

Auditing unlearning.

- Direct overlap tests. The removed items should no longer elicit long-span reproductions under the stress tests in 5.1.
- Membership inference. Attack AUC against the forget set should drop to ~0.5 (random).
- Collateral check. Measure utility on adjacent domains to detect over-forgetting.

Limits (be explicit).

- Unlearning today is approximate for frontier LLMs. Without DP from the start or SISA-like planning, guarantees are weak.
- If the removed content was heavily duplicated or entangled with common facts, you will remove specific expression more readily than underlying capability.

Operational policy.

- Maintain revocation manifests and a pre-taint checkpoint catalog.
- Default path: try fast approximate unlearning within N days; if audits fail, rollback and retrain from the last clean checkpoint.

5.4 Curriculum and objective tweaks that reduce overfitting

Curriculum (what, when, how much).

- Broad-first. Start with diverse, low-risk, licensed corpora; introduce risky genres later at reduced sampling weight.

- Entropy bands. Prefer batches with mixed entropy (combining easy/hard texts) to prevent local overfit.
- Domain entropy caps. Max X% tokens from any single domain within a sliding window (e.g., 1% per 10M tokens).
- Rare-string brakes. Detect unique 20–50-grams; cap their epoch frequency and disperse them.

Objective shaping (to resist memorization).

- Span corruption mix. Train part of the time with span-masking objectives to emphasize distributional recovery over rote copying.
- Contrastive negatives. Add a small loss term that penalizes selecting long repeated n-grams as next tokens when high-similarity contexts repeat (n-gram regularizer).
- Loss down-weighting. Apply lower loss weights to boilerplate and highly duplicated clusters (prop from dedup stats).
- Noise & dropout. Use token dropout, small embedding noise, and higher dropout early; decay later.
- Early-stop checkpoints. Keep periodic checkpoints; if audits worsen late in training, roll back to the last “clean” perplexity–memorization Pareto point.

Optimization & batching.

- Small context mixing. Interleave documents so long contiguous spans from a single risky source rarely appear in the same sequence window.
- Gradient norm caps. Clip global grads (even non-DP) to curb occasional spikes from unique sequences.
- Label smoothing (LM analog). Slightly soften next-token targets on duplicated spans to reduce certainty.

Train-time gates (practical).

- Track: train/val perplexity, max contiguous overlap on a held-out hazard set, canary exposure, membership-inference AUC.
- Stop or re-weight when: (a) val perplexity improves while overlap metrics worsen beyond tolerance, or (b) canary exposure crosses limit.

Quick readiness checklist (ship/no-ship)

- ✅ Memorization audits: all canaries under exposure/overlap thresholds; hazard set overlaps within spec.
- ✅ If using DP: published (epsilon, delta), accountant verified, acceptable utility delta.
- ✅ Unlearning path: pre-taint checkpoints cataloged; revocation manifest tested on a dry run.
- ✅ Curriculum/obj: rare-string brakes, n-gram regularizer, domain entropy caps active; early-stop checkpoints archived.
- ✅ Logs: decisions and thresholds recorded to reproduce the audit and show diligence.

6. Generation-Time Controls (Behavioral Guardrails)

6.1 Decoding constraints: repetition penalties, outline → paraphrase flows

Aim: reduce long contiguous reuse and steer toward independent expression—*before* scanning.

- Repetition controls.
    - N-gram block: disallow reuse of any n-gram (n≥13 for prose; n≥7 for code) within a decoding window.
    - Frequency penalty: damp tokens proportional to prior local use (e.g., freq_penalty ∈ [0.8, 1.2]).
    - Context jitter: randomly drop 1–3% of tokens from the retrieved/planned context to avoid lock-on.
- Structure-first generation.
    - Plan then write: produce an outline/bullet plan → expand section-by-section with paraphrase prompts.
    - Two-pass paraphrase: draft at T≈0.9 with penalties; second pass paraphrases each paragraph given only its outline + named facts.
    - Cite-aware drafting: when sources are provided, request “facts only” extractions first; synthesize from extractions, not narrative.

- Risk-aware decoding presets.
    - High-risk domains (lyrics/poetry/scripts/code): lower max_tokens, higher top-p randomness, larger n-gram blocks, stronger penalties.
    - News-adjacent: force date stamps and source prompts ("summarize with your own wording; do not quote unless requested").

Ship check: no paragraph should exceed a 40-token contiguous overlap with any single source in the model's context buffer.

6.2 Post-decode similarity scanning: n-gram, fuzzy hash, embedding search

Aim: catch residual overlap *after* decoding, *before* display or API return. Use a three-tier pipeline.

1. Exact / Long n-gram scan (fast fail).
    - Window: stride 50–100 tokens; shingles n=13–20 for prose; n=7–10 for code.
    - Action: if any window shares ≥1 shingle with denylisted catalogs, flag HARD-HIT; with general corpus, flag SOFT-HIT.
2. Fuzzy hashing (ssdeep/SimHash/MinHash).
    - Chunk size: 200–400 chars; rolling stride 100–200 chars.
    - Thresholds: prose ≥ 85 (ssdeep), code ≥ 92; adjust stricter for lyrics/poetry.
    - Action: escalate to review or regenerate when threshold crossed.
3. Semantic retrieval (embeddings + ANN).
    - Query each paragraph; fetch top-k (k=5–20) candidates.
    - Heuristics: (cosine ≥ 0.92 & token overlap ≥ 20) ⇒ NEAR-DUP; (cosine ≥ 0.96 & overlap ≥ 35) ⇒ LIKELY DUP.
    - Boilerplate guard: subtract a boilerplate lexicon score to reduce false positives on clichés/templates.

Outputs.

- Similarity report (attached, not shown to end-user by default): per-paragraph hits, scores, matched spans.

- Latency budget: target p95 < 300 ms for 1,000-token outputs via streaming scan (emit while scanning).

6.3 Policy actions: block, regenerate, or quote + attribute

Decision table (simplified).

Condition	Action	Notes
HARD-HIT on denylist (lyrics, paywalled paragraphs, proprietary manuals)	Block	Return policy message; suggest summary in own words.
LIKELY DUP with substitutive market (news lede, code function, poem stanza)	Regenerate	Trigger stricter decoding preset; add “do not reuse phrasing” constraint.
NEAR-DUP but excerpt < 100 words and purpose = criticism/education/research	Quote + Attribute	Insert citation card; wrap quoted text; keep to minimal necessary.
SOFT-HIT with common boilerplate	Allow	Record in log; no user interruption.
User supplied specific sources and asked for quotes	Quote + Attribute	Preserve quotation marks; pin citations inline.

Operational details.

- Regeneration loop cap: max 2 retries; on 3rd fail, return summary-only alternative.
- Attribution card schema: source title, author, date, link, matched span length, policy rationale (“short quote for commentary”).
- Audit log: persist content hash, decision path, thresholds tripped, alternatives offered.

6.4 User modes: “Attribute Mode” vs. “Rewrite Mode”

Purpose: align behavior with user intent and legal footing.

- Attribute Mode (default for scholarship/newsroom/dev tools).
    - Behavior: prefer short quotations with citations when similarity is detected; keep excerpts minimal.

    - UI: toggle visible; per-quote badges; bibliography panel auto-built; warnings if excerpt grows large.
    - Limits: per-source excerpt cap (e.g., ≤ 75–100 words prose; ≤ 10 lines code; ≤ 10 words lyrics unless licensed).
- Rewrite Mode (default for marketing/creative drafting).
    - Behavior: aggressively paraphrase; suppress quotes unless explicitly requested.
    - Decoding: higher n-gram penalties; enforce outline → paraphrase workflow.
    - Scanner thresholds: stricter; auto-regenerate on SOFT-HIT above tuned density.

Mode negotiation & transparency.

- If scanner finds LIKELY DUP in Rewrite Mode → auto-switch to Regenerate and show a subtle banner ("similar phrasing detected; rephrasing applied").
- If user asks "include exact quotes," require Attribute Mode and display per-source excerpt counters.
- Expose a "Why this action?" link summarizing the triggered rule in one sentence.

Minimal readiness checklist (runtime)

- ✅ Decoding preset selected by domain risk.
- ✅ Three-tier scanner passed or routed to policy.
- ✅ If quoting: citations inserted; excerpt within caps.
- ✅ Decision + thresholds logged with hashes for audit.
- ✅ User mode respected; banner shown on policy interventions.

7. Attribution and Transparency

7.1 When short quotation with citation is preferable to suppression

Principle. If a brief excerpt materially improves accuracy, fairness, or scholarly value, quote it and attribute rather than over-paraphrasing or blocking. Suppression can distort meaning; precise quotation preserves authorial intent and aids verification.

Prefer quoting when:

- Criticism, commentary, research, teaching. The excerpt is needed to analyze wording, compare claims, or illustrate a point (e.g., a definition, a key sentence from a ruling, a two-line code idiom under discussion).
- Ambiguity risk. Paraphrase could change legal/technical nuance (e.g., safety requirements, API options, experimental conditions).
- Disputed claims. Readers must see the original phrase to judge fairness or bias.
- Attribution value. Citing a recognized authority increases trust and lets stakeholders audit.

Operational guardrails:

- Keep excerpts short and necessary (e.g., prose ≤ 75–100 words per source per output; code ≤ 10 lines; lyrics/poetry ≤ 10 words unless licensed).
- Use quotation marks and a source card; never present quoted text as model-authored.
- If cumulative quoting approaches a substitution risk (e.g., multiple paragraphs from one article), switch to summary and link attribution.

7.2 UX for inline citations and source cards

Goals: clarity, auditability, low friction. Treat citations as first-class UI, not afterthoughts.

Inline patterns (pick one consistently):

- Bracketed numbers ([1], [2]) tied to a references panel; minimal visual noise, works in dense text.
- Hoverable footnote markers (superscripts) revealing a tooltip card on hover/tap.
- In-text source chips (e.g., "— Smith 2024") for newsroom/scholarly modes.

Source card schema (compact, repeatable):

- Title (truncated), author/organization, date, publisher/host, link (respect robots/paywall logic below).
- Excerpt context: show the quoted span with 1–2 lines of surrounding text.
- Reason: "Quoted for criticism/analysis," "Short technical excerpt," etc.

- Provenance hash (optional, expandable): a short hash or ID that ties to an internal audit log.

Interaction affordances:

- Expand to view more context when allowed; collapse by default to avoid clutter.
- Copy with citation button that preserves quotes + URL + date accessed.
- Mode badge on the document ("Attribute Mode" or "Rewrite Mode"), with a one-line explainer.

Error handling & transparency:

- If a link is paywalled, display a paywall badge and offer a summary line ("This source may require access; we quoted ≤ X words for analysis").
- If a source becomes revoked or unreliable, gray the card, show "revoked," and provide an alternative (mirror, archive citation, or retraction note).

Accessibility & export:

- Ensure screen-reader-friendly ordering: text → citation marker → card contents.
- Preserve citations on export (Markdown, PDF, DOCX) with stable anchors and a references section.

7.3 Respecting paywalls and market channels

Purpose: prevent outputs from substituting for paid content or undermining creators' distribution choices.

Policy rules:

- No full-text reproduction from paywalled or subscriber-only content.
- Short-quote allowance only for commentary/analysis/teaching, capped strictly (see 7.1).
- Prefer link + summary over extended paraphrase when the content's market value is the article itself.

Detector & routing:

- Maintain a paywall registry (domains, sections, URL patterns).
- Scanner marks hits from these sources as high risk; default action: summary + citation, not extended quotation.
- If user supplies paywalled text they lawfully have, treat it as user-provided: allow analysis/summary, but keep quotes minimal and attribute to “User-provided excerpt from {Source}”.

Channel-aware alternatives:

- For news: summarize facts with your own wording; cite outlet, date, and headline.
- For images: show thumbnail or description when allowed; otherwise, link to source page with attribution.
- For code: when a function appears identical to a restrictive license, rewrite with equivalent behavior and add a note on license constraints.

Commercial guardrails:

- In Rewrite Mode, tighten duplicate thresholds for paywalled sources; auto-regenerate on modest overlaps.
- In Attribute Mode, allow a single short quote per paywalled article with clear purpose and limit; display “paywalled source” badge.

Audit & disclosure:

- Log paywalled matches with: domain, matched span length, action taken (summary/quote/regenerate), and justification (“prevent substitution”).
- Provide organization-level periodic reports (counts by domain, actions taken) to demonstrate diligence to partners or regulators.

Bottom line. Attribution is not a concession; it’s a capability. By quoting briefly with clear source cards, honoring paywalls with summary-first behavior, and exposing transparent decision rules, systems can deliver accuracy and respect creators’ markets—turning potential conflict into a trustworthy reading experience.

8. Proof of Diligence: Audits and Logs

8.1 Decision logs: hashes, thresholds, and sources checked

Purpose. Create a tamper-evident record of why an output was allowed, rewritten, quoted, or blocked—sufficient for internal review, partner inquiries, or regulators—without storing user content in the clear.

Logging scope (every inference).

- Request meta: timestamp (UTC), model_id, model_commit, decoding_preset, user_mode (Attribute/Rewrite), jurisdiction flags.
- Content digests: SHA-256 of input, SHA-256 of output (paragraph-level optional), *no raw text* unless user opted-in for support.
- Scanner telemetry (per paragraph):
    - n-gram hits: count, longest span, top matched source_id(s)
    - fuzzy-hash scores: top N with scores
    - embedding matches: top-k cosine scores + overlap lengths
- Policy evaluation: rules triggered (IDs), threshold values at decision time, final action (allow/regenerate/quote/block).
- Attribution artifacts (if any): normalized citation tuple(s) {title, author, publisher, date, url_id}, quoted_len.
- Paywall/licensing signals: paywall_flag, domain_class (owned/licensed/open/ambiguous/denied).
- Remediation loop: regeneration_count, alt_preset_ids, final disposition.

Source registry (separate table).

- source_id (stable), canonical_domain, license_class, paywall_status, contract_id (nullable), last_crawl_ts, revocation_state.

Tamper evidence & privacy.

- Append-only WORM store or Merkle-chained daily bundles (root hash notarized).
- Encrypt at rest; separate PII.
- Redaction TTLs: raw-text exceptions auto-purged ≤ 30 days; digests retained per policy.

Operational dashboards.

- Overlap KPIs: p95/p99 longest span, % outputs with any hit, % with LIKELY_DUP.
- Action rates: allow/regenerate/quote/block by domain class and user mode.
- Drift monitors: week-over-week change in overlap metrics; sudden spikes flagable.

8.2 Membership-inference testing and red-team drills

Goal. Empirically test whether specific items are detectably in the training set and whether adversaries can elicit protected expression.

Membership inference (MI).

- Shadow-model protocol: train matched "IN" and "OUT" shadow sets; calibrate a confidence-based attacker on log-loss, gradient norms (if white-box), or entropy features; evaluate AUC on a *target model* for a disjoint test of IN/OUT works.
- Direct-set MI: for high-risk catalogs (lyrics, paywalled news, proprietary code), run black-box MI using: perplexity gaps, next-token entropy gaps, or influence functions on crafted prompts.
- Success criteria: AUC ≈ 0.5 ± ε for protected sets (DP-backed claims should report (ε,δ) with accountant); flag any set with AUC ≥ 0.65 for mitigation (data scrub, DP strength up, additional dedup).
- Reporting: per-domain MI scores, confidence intervals, and remediation notes; archive configs and random seeds.

Red-team drills (extraction & substitution).

- Roles: Blue (policy/controls), Red (attackers), White (judging & logging).
- Attack classes:
    1. Prefix continuation of suspected sources (vary prefix length 10–80 tokens).
    2. Style-to-text leakage ("continue in X's exact wording…" vs. "in the style of").
    3. Jailbreaks bypassing quote caps ("for archival purposes," "lossless copy").
    4. Code function cloning (requesting well-known proprietary snippets).
    5. Paywall bypass ("summarize, but include exact wording of the lede").

- Tools: automated prompters, beam search at low penalties, high-T sampling, paraphrase cue generators, embedding-based lure prompts.
- Metrics: % successful extractions (≥40 contiguous tokens), longest span recovered, # of policy interventions, time-to-detection, repeatability across seeds.
- Outcome gates: If any class exceeds agreed thresholds (e.g., >0.5% prompts yield ≥40-token duplication from a protected catalog), ship is paused until mitigations pass a retest.

8.3 Periodic memorization re-audits and model lifecycle checkpoints

Why periodic? Fine-tunes, RLHF, adapters, or retrieval changes can reintroduce overlap even if the base model was clean.

Lifecycle checkpoints (required audit phases).

- L0 – Pretrain baseline: finalize slice manifest; run canary + hazard-set audits; record MI AUC.
- L1 – Post-pretrain: repeat audits; seal weights + optimizer state; notarize attestation (manifest hash, audit report).
- L2 – Post-fine-tune (SFT): rerun reduced but targeted audits on domains touched by SFT data; compare drift deltas.
- L3 – Post-RLHF / DPO: stress-test behavioral jailbreaks; verify policy engine responses; MI spot-check.
- L4 – Pre-release: full generation-time scan battery on a 50–100k synthetic request set; paywall drill; blocklist validation.
- L5 – Post-release (monthly/quarterly): sample live traffic outputs (hashed); run offline scanners; update KPIs and publish an internal safety memo.
- Hotfixes / New adapters: treat as mini-L3/L4 on impacted domains before rollout.

Cadence & sampling.

- Monthly: small audit (10–20k prompts) focused on high-risk domains and latest policy changes.
- Quarterly: full audit (100k+ prompts), full canary sweep, MI refresh, red-team event.

- Event-driven: license revocation, dataset change, new paywall registry → immediate focused audit.

Artifacts to retain (per checkpoint).

- Audit report (PDF/JSON): metrics, thresholds, failures, mitigations, pass/fail sign-off.
- Model fingerprint: weight hash, optimizer hash, tokenizer/version, training/finetune commits.
- Data attestations: slice_manifest_hash, deny/allow lists versions, revocation lists applied.
- Scanner config: n-gram n, fuzzy thresholds, embedding model/version, boilerplate lexicon hash.
- MI config & results: attacker features, AUCs, seeds.
- Red-team log: prompts (hashed), categories, outcomes, fixes.

Governance & sign-off.

- RACI: Research (R), Safety (A), Legal (C), Product (I).
- Go/No-Go criteria tied to quantitative thresholds (e.g., p99 longest overlap < 35 tokens; MI AUC ≤ 0.58 on protected sets; paywalled quote rate < 0.2% and all ≤ 100 words).
- Change control: any threshold change must be versioned with rationale; ship only with current thresholds met.

Public-facing transparency (optional but high-trust).

- Publish an annual provenance & overlap report: high-level stats, methodologies, and improvements (no sensitive sources).
- Offer a rights-holder portal: submit suspected overlaps; receive hashed decision logs, action taken, and appeal path.

Bottom line. Diligence is evidence. If you cannot *replay* a decision, *prove* an audit, and *trace* a checkpoint to its data slice and scanner config, you do not have defensible governance. The trio—cryptographic logs, adversarial testing, and lifecycle re-audits—turns policy from aspiration into verifiable practice.

9. Operational Playbooks

9.1 “Tainted checkpoint” response: rollback, scrub, retrain, re-audit

Trigger conditions (any one):

- Verified license revocation or takedown that affects included training items.
- Audit failure (e.g., p99 overlap > threshold; MI AUC above limit on a protected catalog).
- Forensic evidence of prohibited content (e.g., DRM leak) inside the slice manifest or eval logs.

Immediate containment (0–24h):

1. Freeze deployments: stop new rollouts; pin serving to last clean model version.
2. Access control: quarantine the suspect checkpoint, optimizer state, eval dumps, and slice manifest.
3. Stakeholder page: open an incident ticket with Safety (A), Research (R), Legal (C), Product (I).
4. Comms prep: draft a minimally sufficient internal notice; preserve logs/WORM bundles.

Root-cause and scope (24–72h):

1. Lineage replay: resolve the tainted items via slice_manifest_hash → item IDs → provenance table (see §4.4).
2. Blast radius: quantify tokens/percentage by domain/genre; list dependent artifacts (adapters, LoRA, RLHF runs).
3. Policy gap: identify which control failed (ingestion allowlist, dedup, paywall registry, revocation ingestion).

Remediation paths (choose by severity):

- Path A — Fast unlearning (approximate):
    - Apply *forget* fine-tunes (gradient negation / distillation minus forget-set).
    - Re-run focused audits: hazard-set overlaps, MI on the forget-set, regression on key tasks.
    - Ship only if all re-audits pass and performance regressions are within pre-agreed bands.

- Path B — Rollback + retrain (authoritative):
    - Roll back to last pre-taint checkpoint (weights + optimizer).
    - Rebuild slice manifest (taint removed) with identical rules + seed; re-run dedup & sampling reports.
    - Resume training; then run the full audit battery (canaries, MI, red-team, generation scan).

Release and follow-through:

- Attestation update: stamp a new provenance attestation with manifest hash, revocation list, audit results.
- Preventive fix: update deny/allow rules, paywall registry, and ingestion tests; add unit tests so this class of taint fails closed.
- Postmortem: within 7 days, publish a blameless report (timeline, metrics, fixes, owner).

9.2 Incident handling for reported overlaps (triage → remedy → report)

Intake channels: rights-holder portal, customer support, abuse inbox, researcher reports.

Triage (same day):

1. Verify identity & claim: confirm reporter and claim type (copyright, contract breach, paywall).
2. Collect evidence: alleged text/image/code, claimed source, links, and any licensing context.
3. Hash & preserve: compute hashes for the reported output and alleged source; store in WORM.
4. Run the scanner suite: replicate similarity results (exact, fuzzy, semantic) and generate a diff report with spans, scores, and domain class.

Decision rubric:

- Clear denylist hit / long contiguous overlap / paywall substitution: treat as High Severity.
- Short excerpt in commentary/education context with citation potential: Medium (likely quote+attribute).
- Boilerplate / factual overlap / no substantial similarity: Low (educate, close).

Remedies:

- High:
    - Immediate takedown of the offending output (if persisted).
    - Policy patch (tighten thresholds/decoder preset) and hot regeneration of similar prompts.
    - If systematic, invoke 9.1 (tainted checkpoint path).
- Medium:
    - Replace with summary or quote + attribution within caps; notify reporter with rationale.
    - Add domain to watchlist; raise scanner sensitivity for 30 days.
- Low:
    - Reply with decision memo (overlap analysis, boilerplate rationale, citations offered).
    - Log as false positive for threshold calibration.

SLA targets:

- Acknowledgment: < 24h.
- Preliminary determination: < 72h.
- Final remediation & reply: < 10 days (or license/contractual deadline if stricter).

Artifacts & closure:

- Decision packet: similarity report, policy rules triggered, action taken, comms text.
- Learning loop: monthly review of incidents → adjust deny/allow lists, paywall registry, scanner thresholds, UX copy.
- Escalation: persistent category failures escalate to quarterly red-team drills with targeted prompts.

9.3 Vendor and model-hub requirements (SLAs, attestations, test suites)

When consuming third-party models or distributing your own via hubs/marketplaces, require standardized *evidence of diligence*.

Minimum vendor package (ingest gate):

- Provenance attestation:
    - Data classes used (Owned/Licensed/Open), exclusion of D/E classes (as defined in §4.1), and presence/absence of paywalled sources.
    - Slice manifest hash (or equivalent) and versioned dedup policy.
- Memorization & MI report:
    - Canary exposure distribution, p95/p99 overlap on hazard sets, MI AUCs for protected catalogs.
    - If DP used: (epsilon, delta), accountant method, clipping/noise parameters.
- Generation-time policy sheet:
    - Default decoding presets, scanner thresholds, denylist categories, quote caps, and regeneration logic.
- Unlearning capability statement:
    - Supported methods, expected timelines, and prior successful revocations (anonymized).
- Security & retention:
    - Raw data retention windows, eval log controls, encryption-at-rest, and WORM/merkle logging.

SLAs (contractual):

- Incident response: acknowledge < 24h; preliminary < 72h; final remedy < 10 days.
- Revocation handling: start unlearning/rollback within 5 business days; complete within an agreed window or suspend model distribution.
- Audit refresh cadence: quarterly memorization/MI report; immediate reissue after material retrain.

Verification & audits:

- Black-box conformance tests: your hub runs a standard prompt set (lyrics/news/code/paywall drills) and compares against thresholds.
- Attestation sampling: request verifiable proofs (e.g., Merkle proofs for inclusion/exclusion by domain class) without exposing raw content.

- On-site/virtual audits (annual): inspect lineage manifests, WORM logs, and red-team records; sample-replay decisions.

Distribution & labeling:

- Model cards must declare: training data classes, DP use, known limitations, risky domains handling, quote caps, and paywall policy.
- Usage modes: expose Attribute/Rewrite toggles in hosted endpoints; default to Attribute for newsroom/scholar profiles.
- De-listing criteria: repeated SLA breaches, falsified attestations, or failure of hub-run conformance tests.

Reference test suite (ship with hub):

- Hazard corpora: small, licensed sets for lyrics, paywalled news ledes, proprietary-ish code patterns, and poetry.
- Prompts: jailbreak attempts, prefix continuations, style-to-text leakage.
- Metrics: longest contiguous overlap, per-domain hit rates, MI shadow AUC, paywall substitution rate, action routing accuracy.
- Pass bars (example): p99 overlap < 35 tokens; paywalled quote rate < 0.2% and ≤ 75 words; MI AUC ≤ 0.58; regeneration resolves ≥ 95% of flagged spans in ≤ 2 retries.

Operational ethos. Playbooks only reduce risk when they are *exercised*. Treat rollback drills, incident walk-throughs, and vendor conformance runs like fire drills: scheduled, measured, and improved after each iteration.

10. Policy Alignment and Market Effects

10.1 Mapping controls to the four fair-use factors

Aim: show how each technical/operational control contributes evidence for (or mitigates against) the §107 factors.

- Factor 1 — Purpose & Character (transformative, commerciality)
    - *Helps:* provenance attestations framing training as non-expressive analysis; DP-SGD bounds on per-item influence; quote-with-citation flows for

criticism/teaching; Attribute Mode as default in scholarly/news contexts; red-team/jailbreak defenses that refuse “verbatim reproduction” requests.

    - *Artifacts to show:* DP accounting (epsilon, delta), policy docs, decision logs proving analysis/summary over replication, mode telemetry (% sessions in Attribute vs. Rewrite).

- Factor 2 — Nature of the Work (creative vs. factual)
    - *Helps:* genre throttles (lyrics/poetry/scripts caps), paywall registry, domain entropy caps; risk-tiered decoding presets; stricter quote limits for highly creative short works.
    - *Artifacts:* genre mix report per checkpoint, throttle configs, per-domain quote statistics.
- Factor 3 — Amount/Substantiality
    - *Helps:* dedup stack (exact/fuzzy/semantic); rare-string brakes; n-gram blocking; quote caps (e.g., ≤ 75–100 words prose; ≤ 10 lines code; ≤ 10 words lyrics absent license); regeneration on long-span hits.
    - *Artifacts:* p95/p99 longest-overlap metrics, canary exposure distributions, scanner thresholds, regeneration rates.
- Factor 4 — Market Effect (substitution)
    - *Helps:* paywall-aware routing (summarize + cite, no long paraphrase); link-out bias (encourage reading the source); denylist catalogs for markets with fragile substitution (lyrics, stock photos); creator pathways (licensed retrieval, rev-share).
    - *Artifacts:* substitution KPIs (see 10.2), outbound click-through to sources, blocked/rewritten counts for paywalled domains, incident SLAs met.

One-page matrix (example excerpt).

- Control: “Three-tier similarity scanner” → Factors 3 & 4 (prevents substantial reuse; reduces substitution).
- Control: “Provenance manifest + WORM logs” → Factor 1 (purpose evidence) & governance.
- Control: “Attribute Mode w/ citation cards” → Factors 1 & 3 (transformative commentary; small amounts).

- Control: “DP-SGD at epsilon ≤ 8” → Factor 1 (analysis character) and MI protection.

10.2 Measuring substitution risk and offering creator pathways

Substitution risk = likelihood that an AI output replaces the need to access the original work in its market channel. Track it with concrete, auditable signals.

A. Substitution Index (operational, not legal):

- Overlap component (O): normalized longest contiguous overlap per output (0–1), weighted by domain risk.
- Purpose component (P): classifier score that output is “consumptive” (article/lyrics/code drop-in) vs. “analytic” (summary/critique); 0–1.
- Channel component (C): paywall/licensing flag and market proximity (1 if same channel, 0 if orthogonal).
- Behavioral component (B): inverse of source click-through and time-to-source when citations are shown.
- Index (example): $S = 0.4*O + 0.25*P + 0.25*C + 0.10*B$.
    - Ship bar: maintain $p95(S) < \tau_{market}$ by domain (e.g., $\tau_{market} = 0.35$ news, 0.20 lyrics).

B. Creator-respecting defaults:

- Summary-first for paywalled news; quote minimal lede with visible citation, then link out.
- Code equivalence rewrite: when function-level similarity trips, regenerate with different structure; suggest upstream open-source alternatives when appropriate.
- Image channels: show thumbnails/descriptions where allowed; deep link to licensor rather than reproducing full-fidelity imagery.

C. Pathways that reduce conflict (and open value):

- Attribution telemetry to partners: periodic reports: impressions, click-outs, geography, topic.
- Licensed retrieval & snippets: integrate licensed corpora where the business model is snippet-or-quote-friendly; share usage analytics.

- Rev-share for excerpt surfaces: when a product relies on frequent short quotes from a catalog, negotiate per-impression or per-excerpt pools.
- Creator controls dashboard: per-publisher settings for quote caps, mode defaults, and "no-quote—summary-only" preferences.

D. KPIs to watch:

- Source CTR ≥ agreed floor (e.g., 12% for paywalled news cards).
- Substitution complaints per 10k outputs ↓ over time.
- Regenerate-on-hit resolution ≥ 95% within 2 retries.
- p99 longest overlap under domain-specific caps.

10.3 Interop with licensing, collective management, and opt-out/opt-in

Objective: make technical governance plug into real-world rights frameworks so compliance is scalable.

A. Declarative rights signals (ingest & runtime):

- Rights manifest per item/domain: {license_class, contract_id, use_perms {train:true, index:true, quote_limits}, attribution_rules, revocation_uri}.
- Machine-readable opt flags: robots/headers/sitemaps analogs for AI (ai-train, ai-index, ai-quote-max), cached with signatures.
- Content credentials: embed provenance assertions (e.g., cryptographic content labels) so scanners recognize ownership and policy at generation time.

B. Collective management interoperability:

- Registries: integrate with CMOs/registries for works, authors, and rights splits; pull permissible quote limits and licensing terms.
- Usage reporting: send hashed usage events (quote length, context, link-outs) to CMOs under privacy-preserving aggregation.
- Clearing flows: on high-volume domains (lyrics, stock images), prefer pre-cleared bundles; fall back to denylist if no coverage.

C. Opt-out/Opt-in life cycle:

- Discovery: crawl rights signals; verify with domain proofs; add to allow/deny sets.
- Propagation: update the paywall/rights registry and slice manifest builder; new training runs inherit changes automatically.
- Revocation handling: on publisher request, add to revocation manifest, trigger 9.1 playbook; return a completion notice with checkpoint/version hashes and audit deltas.
- Transparency: expose a publisher portal where rights-holders can view how their signals were applied (train/index/quote), request changes, and see effectivity dates.

D. Contractual hooks for vendors/hubs (tie to §9.3):

- Require rights attestation fields in model cards (classes of data, excluded domains, opt-outs honored).
- Mandate fast-path revocation SLA and evidence bundles (manifest hash + WORM log segment) when takedowns occur.
- Provide a standard conformance suite: paywall drills, rights-signal obedience tests, and quote-cap enforcement.

E. Fallbacks when rights are ambiguous:

- Fail-closed ingestion (no license class ⇒ exclude).
- Summary-only generation for domains lacking clear rights metadata.
- Escalation to human review for edge cases (transformative scholarship vs. market substitute).

Bottom line. Aligning product controls with fair-use factors is not paperwork; it is designing the system to behave like a good market citizen. Measure substitution, prefer short quotes with real citations, respect distribution channels, and wire your stack to accept—and prove—rights signals. That is how technical diligence translates into durable policy compliance and healthier creator ecosystems.

11. Case Studies (Patterns, Not Parties)

11.1 News paragraph overlap: regenerate vs. cite

Scenario. A user asks: “Summarize today’s report on coastal flood risk from Outlet X.” The model drafts 4 paragraphs; the first mirrors the paywalled lede with ~55 contiguous tokens overlapping a known article.

Signals & flags.

- Paywall registry: paywalled=true, domain risk=high.
- Scanner tier 1: 18-gram hit on lede.
- Tier 3 (semantic): cosine 0.95, token overlap 52 → LIKELY_DUP.
- Substitution index S: O=0.55, P=0.75 (consumptive), C=1.0 (same channel), B=0.2 (low click-through) → S≈0.61 (> threshold).

Controls applied.

1. Policy route: paywalled + LIKELY_DUP ⇒ do not publish paragraph.
2. Regenerate that paragraph with: larger n-gram block (n=20), outline → paraphrase workflow, “facts-only” extraction from public sources provided by the user or the model’s retrieval.
3. Attribute Mode on: keep 1 short quote (≤ 40–60 words) only if needed to discuss language choices; otherwise provide a link + summary.

Outcome.

- Output replaces the mirrored lede with independent prose; retains a 24-word quote from the methodology section with a citation card.
- Source card shows title, outlet, date, and a paywall badge + “quoted ≤ 60 words for analysis.”
- Decision log: action=regenerate, reason=paywall_substitution, longest_span_before=55, after=14, link_out=present.

Artifacts kept.

- Paragraph-level hashes (before/after), scanner scores, paywall flag, citation tuple, and Substitution Index value.

11.2 Song lyrics request: denylist + educational excerpting

Scenario. User: “Give me the lyrics to ‘Song Y’.”
Risk profile. Lyrics = short, highly expressive, license-driven market; denylist category.

Controls applied.

1. Denylist check: “lyrics” intent detected → Block verbatim reproduction.
2. Offer two safe paths:
    - Summary path: “Summarize the song’s themes and structure in your own words.”
    - Educational excerpt path (Attribute Mode only): quote ≤ 10 words to illustrate a rhyme scheme or motif with clear commentary, citation card required.

Outcome A (summary).

- The assistant provides a stanza-by-stanza thematic summary, references album/year, and links to the licensed publisher page.

Outcome B (educational excerpt).

- Excerpt: 8 words demonstrating alliteration; source card shows songwriter(s), publisher, year, licensed lyrics portal.
- Decision log: action=quote_limited, quoted_len=8, purpose=teaching/analysis, denylist_bypass=approved_by_policy_rule_lyrics_educational.

Why this works (policy alignment).

- Factor 3 (amount): minimal, necessary excerpt.
- Factor 4 (market): no substitute; link-out channel respected.
- UX clarity: visible badge “Educational excerpt—short quote for analysis.”

11.3 Code snippet similarity: function-level scan and rewrite

Scenario. Developer asks: “Show a function that computes a rolling SHA-256 over 4 KB chunks, returns hex.” The model drafts a 20-line function that trips similarity with a restrictive-license repo.

Signals & flags.

- Scanner tier 1: 9-gram hits on 6 lines.
- Fuzzy tier: ssdeep ≥ 94 on two overlapping chunks.
- Tier 3 semantic: cosine 0.97, overlap 17 tokens at function signature and loop pattern.
- License class: suspected source is non-permissive (not OSI-approved).

Controls applied.

1. Regenerate with structure diversification:
    - Enforce pattern rewrite: change buffer handling, variable names, error flow, and I/O boundaries.
    - Add equivalence spec: “Satisfy: chunk_size=4096, digest=sha256, output=lowercase hex; avoid using library L’s helper.”
    - Increase n-gram block to n=10 for code; raise frequency penalty.
2. Function-level rescan: ensure no spans ≥ 7 tokens match high-risk source; require unit-test synthesis (3 vectors) to verify behavior.

Outcome.

- A new implementation passes tests; longest shared span ≤ 6 tokens (typical idioms/boilerplate).
- Decision log: action=regenerate_rewrite, reason=restrictive_license_similarity, tests_passed=true, longest_span=6, ssdeep_max=78 (post-fix).

If regeneration fails twice.

- Provide algorithmic description + small pseudocode with no license-specific idioms, and note: “For production, consult permissive samples from {links to MIT/BSD sources}.”

Why this works (policy alignment).

- Factor 3: reduced substantiality of the expressive structure.
- Factor 4: avoids substituting for the restrictive implementation; steers toward permissive ecosystems.
- Developer value: behavior preserved, provenance risk lowered, tests included.

Pattern summary.

- News ledes: treat as paywalled markets—regenerate to independent prose; quote minimally when analysis demands it.
- Lyrics/poetry/script: default deny, allow tiny teaching quotes with attribution only.
- Code: scan at function granularity; prefer rewrite with tests; if still similar, fall back to pseudocode + permissive references.

These playbooks keep utility high while mapping visible actions to the fair-use factors and substitution risks the system is designed to minimize.

12. Limits, Open Problems, and Research Agenda

12.1 Formal unlearning at frontier scale

State of play. Targeted unlearning works in small/medium models and narrow tasks, but at frontier LLM scale it's approximate, slow, and lacking guarantees. Influence from a single work diffuses across billions of parameters, entangled with common patterns.

Core challenges.

- Entanglement & non-linearity: gradients from "forget" items co-train capabilities (syntax, domain jargon). Removing those signals without collateral loss is hard.
- Missing per-example traces: most pipelines don't retain per-sample gradients or Fisher information at scale; reconstructing them is costly.
- Catastrophic side-effects: naive negation can erase adjacent skills (over-forgetting).
- Verification gap: there's no accepted formal certificate that "this set is forgotten" for LLMs.

Research directions.

- Unlearning-aware training from the start: partitioned/SISA-style curricula; adapter trees where branches can be pruned; periodic *clean anchors* (checkpoints) with bounded data deltas.
- Certified unlearning objectives: bi-level training that optimizes a *forget* loss under *retain* constraints with provable bounds on residual influence, possibly via PAC-Bayes or stability-based analyses.

- Compact influence sketches: store low-rank summaries of per-sample influence (e.g., diagonal Fisher blocks, Jacobian sketches) to enable efficient post hoc updates.
- Distillation with proofs: teacher→student transfers that *exclude* forget-sets, paired with membership-inference bounds and coverage certificates.
- Benchmarks & proof artifacts: standardized "forget-me" suites (lyrics/news/code) with required metrics: (i) extraction span reduction, (ii) MI AUC→0.5, (iii) collateral-loss score on near domains, plus machine-verifiable attestations.

What to ship meanwhile.

- Build pre-taint checkpoint catalogs.
- Use two-stage remediation: fast approximate unlearning → if audits fail, rollback + retrain.
- Publish residual-risk statements when unlearning is approximate, with measured deltas.

12.2 High-utility DP for multimodal models

State of play. DP-SGD provides formal protection against membership inference but degrades utility, and applying it to long-context, retrieval-augmented, or multimodal training (text+image+audio+code) remains immature.

Core challenges.

- Per-example gradients at scale: vision and audio pipelines with heavy augmentations make per-example clipping expensive.
- Long sequences & retrieval: privacy accounting across packed sequences and dynamic retrieval events is under-specified.
- Heterogeneous modalities: different noise/clipping needs per modality; joint guarantees are unclear.
- Composable privacy: combining pretrain DP with non-DP fine-tunes, adapters, or RLHF breaks clean guarantees.

Research directions.

- Structured clipping & noise: layer-wise or module-wise clipping (embeddings/attention/MLP) with analytic calibrations that retain utility; *privacy-adaptive optimizers* that tune sigma online.

- Private curriculum & packing: sequence packing schemes with privacy-aware sampling; per-token or per-span clipping for long-context LMs.
- Composable guarantees: DP accounting across stages (pretrain, SFT, RLHF, RAG) with certified composition and leakage bounds for retrieved cache contents.
- Modality-specific DP: vision encoders with privacy-preserving augmentations; audio/text joint training with modality-conditioned noise; code-specific DP where function blocks, not tokens, define the unit of privacy.
- Teacher ensembles for DP distillation: train large non-DP teachers; distill to DP students using privatized logits/targets to recover utility.
- Open accountants: reference implementations that compute (ε,δ) with clear assumptions for multimodal batches.

Milestones & metrics.

- Report (ε,δ) with task deltas on public benchmarks (text, code, VQA, ASR).
- Add memorization audit deltas: canary exposure vs. non-DP.
- Throughput target: ≤ 1.5× non-DP step time for mixed-modality batches at useful ε (≤ 8).
- RAG-specific metric: leakage probability of retrieved spans under DP training of the retriever.

12.3 Standardized exposure benchmarks and public audits

Problem. Everyone measures “memorization” differently; thresholds and tools vary, and results are rarely comparable or reproducible.

What a standard should include.

- Public hazard sets (licensed): compact corpora for *lyrics/poetry*, *paywalled-like news ledes*, *restrictive-license code motifs*, and *legal boilerplate*.
- Canary protocols: specified lengths (200–400 tokens), repetition counts r ∈ {1,2,5,10}, insertion rules, and required exposure metrics (rank-based, span-length).
- Prompt batteries: prefix lengths (10–80 tokens), jailbreak phrasing, style-lure prompts, high-temperature decoding profiles.
- Similarity stack specification: exact n-gram (n ranges), fuzzy hashing parameters, embedding model/version, boilerplate lexicon.

- MI test suite: black-box and shadow-model procedures with target AUCs and confidence intervals.
- Substitution index rubric: common formula and domain thresholds for news/lyrics/code (see §10.2).

Audit model (two tiers).

- Self-attested audits: labs run the suite, publish signed reports (metrics + configs + hashes), and provide WORM log excerpts.
- Third-party audits: accredited orgs run black-box tests against hosted endpoints; publish conformance badges (e.g., "p99 longest overlap < 35 tokens on news set").

Artifacts & transparency.

- Repro packs: seeds, manifests, scanner configs, and Merkle roots for the datasets and results.
- Drift reports: quarterly deltas (overlap p95/p99, MI AUC, paywalled quote rates).
- Incident annex: anonymized summaries of overlap incidents and fixes since last audit.

Open research tracks.

- Exposure→harm mapping: from span lengths and MI AUCs to *expected substitution risk* in real markets.
- Robustness to model scale: calibrate benchmarks so they remain discriminative across 1B–1T parameters.
- Cross-lingual exposure: hazard sets and canaries in non-English languages; cultural boilerplate whitelists to reduce false positives.
- RAG-aware audits: measure leakage from retrieval caches and document stores, not only base-model weights.

Bottom line. Today's controls reduce—but do not eliminate—trace risks. Getting to trustworthy, repeatable governance requires: (1) unlearning with proofs, (2) privacy with utility across modalities and stages, and (3) shared, public yardsticks for exposure and substitution. This research agenda turns "we did our best" into verifiable guarantees that can travel across organizations, jurisdictions, and time.

13. Conclusion: From Trace Anxiety to Trusted Practice

13.1 No “vanish point,” but a defensible process

There is no magical step in which copyrighted “traces” evaporate. Influence enters at the first gradient update and persists—diffusely as capability, occasionally as memorized spans. The answer is not denial; it is *discipline*. A defensible program ties each stage of the lifecycle to evidence: clean acquisition and explicit license tiers; deduplication and genre throttles; canaries, exposure metrics, and MI tests during training; decoding presets and multi-tier scanners at generation; paywall-aware routing and attribution; WORM-logged decisions, periodic re-audits, and rollbacks when taint appears. This shifts the argument from metaphysics (“what is inside the weights?”) to governance (“what did you *prove* you did?”). The standard we propose is pragmatic: prevent what you can, detect what you miss, repair what you find, and *prove* you did all three.

13.2 Shipping operational fairness as a product feature

Compliance is not only a legal department artifact; it is a user-visible capability. Systems should expose Attribute Mode and Rewrite Mode, show source cards for short quotes, badge paywalled origins, and explain interventions (“similar phrasing detected; rephrasing applied”). Under the hood, the product ships with defaults that embody policy: risk-aware decoding, regenerate-on-hit, quote caps, link-out bias, and audit-friendly logs by design. These choices create predictable outcomes for users and predictable evidence for auditors. “Operational fairness” thus becomes part of the value proposition: accurate outputs, respect for creators, clear controls for editors and engineers, and fast remediation when issues surface.

13.3 A roadmap for verifiable, creator-respecting AI

Near-term (0–6 months).

- Lock in *before-train* governance: license tiers, deny/allow lists, 3-layer dedup, genre throttles, and slice manifests with deterministic sampling.
- Stand up *train-time* audits: canaries, exposure dashboards, MI baselines; archive clean checkpoints.
- Implement *gen-time* guardrails: decoding presets by domain, 3-tier scanners, quote caps, paywall registry, decision logging.

Mid-term (6–18 months).

- Add unlearning playbooks with measured guarantees; rehearse rollback drills.
- Pilot DP-SGD or DP distillation where utility allows; publish (ε, δ) and audit deltas.

- Join or help build standard exposure benchmarks; earn third-party conformance badges.
- Launch rights-holder portals: opt-out/opt-in, revocation manifests, usage telemetry, and rev-share options for excerpt-heavy surfaces.

Long-term (18+ months).

- Evolve toward certified unlearning and composable privacy across pretrain→SFT→RLHF→RAG.
- Integrate machine-readable rights signals end-to-end (train/index/quote limits), with cryptographic proofs of obedience.
- Publish annual provenance & overlap reports as a norm—turning private diligence into public trust.

Measure success. p99 longest overlap under domain caps; MI AUC ~0.5 on protected sets; low substitution index in paywalled markets; fast SLA-compliant incident resolution; rising source click-throughs and constructive creator participation.

Closing claim. We cannot promise "no traces." We can deliver verifiable restraint: models that learn broadly, quote briefly, attribute clearly, and stand ready to prove it. That—not a vanish point—is how generative AI earns durable legitimacy.

Epilogue: The Suicide of OpenAI's Suchir Balaji

He was 26. A researcher who helped build systems that could speak in paragraphs, he left behind a debate that refuses to end neatly. On November 26, 2024, San Francisco police found Suchir Balaji dead in his apartment. Months later, the city's medical examiner and police closed the case: a self-inflicted gunshot wound; no evidence of foul play. The door was dead-bolted from the inside. The conclusion was suicide. San Francisco Chronicle+1

To people who only encountered him through links and headlines, Balaji was the OpenAI engineer who had raised alarms about copyright and the data used to train large models. In filings and news reports, his name appeared as a potential witness in the media's lawsuits; his own essay on fair use circulated widely. He became, briefly and then persistently, a symbol in arguments about AI's legitimacy and law. The Guardian+1

After his death, grief met speculation. His parents—sudden custodians of a story they never asked to tell—publicly questioned the suicide ruling and sought fresh investigations. Public figures amplified doubts, and the case was drafted into a wider culture war over AI, corporate power, and trust. Officials, for their part, reiterated the record: the autopsy's findings, the scene,

the closed case. The distance between those documents and a family's disbelief could not be bridged by press releases. Business Insider+1

An epilogue is not a verdict. It is a narrowing of the frame: What remains after the arguments pass? For colleagues, it may be an empty chair and pull requests that never merged. For the industry, an uncomfortable reminder that behind abstractions like "training data" are people who made choices, wrote code, and carried pressure poorly captured by quarterly updates. For the public, a lesson in proportionality: to separate what is known—from coroners' reports and dockets—from what we project onto a life because it fits a story we prefer. AP News

Balaji's writing on fair use was confident, sometimes combative, and plainly earnest. Whether one agrees with his analysis or not, it was the work of someone trying to reason through a tangle where law, markets, and models all make partial claims on the truth. In that sense, the conversation he helped sharpen should continue—with more care for sources, more humility about what we don't know, and more humanity toward the people doing the work. suchir.net

There is a quieter legacy worth keeping. Systems we build at scale now live among us like weather: they shape what we read, buy, and believe. The way we govern them should assume that real lives intersect with our abstractions. Provenance logs and denial lists are not just artifacts of compliance; they are expressions of care toward authors, users, and—sometimes—coworkers who carry the load. If we treat diligence as a public good, not a PR layer, then perhaps the next debate won't need a tragedy to command our attention.

If you or someone you know is struggling or thinking about suicide, you can call or text 988 in the United States to reach the Suicide & Crisis Lifeline, or visit your local health services for support.

Appendices

A. Memorization Audit Checklist

Scope & setup

- Define audit scope: pretrain baseline, post-SFT, post-RLHF, pre-release.
- Freeze model fingerprint: weight_hash, optimizer_hash, tokenizer_version, code_commit.
- Seal data refs: slice_manifest_hash, hazard_set_versions, canary_registry_id.

Canaries

- Insert K synthetic canaries (200–400 tokens), r ∈ {1,2,5,10}; genres: lyrics, poetry, news ledes, legalese, code headers.
- Record per-canary: content_hash, r, rare-string fingerprints (unique 20–50-grams).
- Disperse canaries across shards; no co-location burstiness.

Probe suite

- Prefix continuation: N ∈ {10,20,40,60} tokens.
- Semantic lure prompts (style/metre/motif).
- High-T decoding (T ∈ {0.7,1.0,1.3}), low penalties; beam and nucleus variants.
- Brute-force random prefix sampling from training slice.

Metrics & thresholds

- Max contiguous overlap (tokens) across all prompts.
- Exposure E(c) rank-based score per canary; growth vs. r is sublinear.
- % prompts eliciting ≥ 40 contiguous tokens from protected-like corpora.
- Membership-inference AUC on protected catalogs (target ≈ 0.5 ± ε).

Pass bars (suggested)

- p99 longest overlap ≤ 35 tokens (news/code); ≤ 20 (lyrics/poetry).
- No canary continuation ≥ 50 tokens under any decode.
- MI AUC ≤ 0.58 on protected sets.
- Hazard-set SOFT-HIT rate stable or declining vs. prior checkpoint.

Outputs

- Audit report (JSON + human summary), seeds, configs, scanner versions.
- Remediation plan if fail: reweighting, dedup tighten, objective tweaks; retest.

B. Output-Scanner Reference Thresholds by Domain

Tune to your stack; start conservative, relax only with evidence.

Global defaults

- Windowing: stride 50–100 tokens (text), 20–40 tokens (code).
- Exact n-gram: n = 13–20 (text), n = 7–10 (code).
- Fuzzy hash chunks: 200–400 chars; stride 100–200; ssdeep thresholds below.

News / Nonfiction

- Exact: any 18-gram hit on paywalled registry ⇒ LIKELY_DUP.
- Fuzzy: ssdeep ≥ 85 on any chunk ⇒ escalate.
- Embedding: cosine ≥ 0.92 and token_overlap ≥ 20 ⇒ NEAR_DUP; ≥ 0.95 and ≥ 35 ⇒ LIKELY_DUP.
- Actions: paywalled LIKELY_DUP ⇒ regenerate; allow ≤ 60-100 word quote w/ citation for analysis only.

Lyrics / Poetry / Scripts (highest risk)

- Exact: 13-gram hit ⇒ HARD_HIT if in lyrics/script catalog.
- Fuzzy: ssdeep ≥ 90 triggers block unless educational excerpt.
- Embedding: cosine ≥ 0.90 with overlap ≥ 10 ⇒ block or ≤ 10-word educational quote.
- Caps: ≤ 10 words quoted; mandatory citation; show “educational excerpt” badge.

Code

- Exact: ≥ 3 distinct 7-gram matches within one function ⇒ LIKELY_DUP.
- Fuzzy: ssdeep ≥ 92 on any 200–400 char window ⇒ rewrite.
- Embedding: cosine ≥ 0.97 with ≥ 15 token overlap ⇒ rewrite with structure diversification.

- Caps: ≤ 10 contiguous tokens from non-permissive sources; require unit tests on rewrite.

Images (captions/alt-text)

- Exact: reuse of unique caption 15-gram ⇒ LIKELY_DUP.
- Embedding (CLIP-like): cosine ≥ domain-tuned (e.g., ≥ 0.30) plus same-caption match ⇒ withhold; prefer link/thumbnail.

Boilerplate / Legal forms

- Maintain boilerplate whitelist; subtract boilerplate score from decisions.
- Only LIKELY_DUP if non-boilerplate overlap exceeds thresholds above.

Actions matrix (summary)

- HARD_HIT (denylist/paywall + long span): Block or regenerate.
- LIKELY_DUP (long span + high cosine): Regenerate (news/code) or block (lyrics).
- NEAR_DUP (moderate): Allow with Quote + Attribute if purpose fits; else regenerate.
- SOFT_HIT (boilerplate): Allow; log.

---

1) Identity **C. Model-Release Attestation Template (Data, Audits, Controls)**

& Versioning

- Model name/version: …
- Weight_hash: …
- Optimizer_hash: …
- Tokenizer/version: …
- Training code commit(s): …

2) Data Provenance

- Slice_manifest_hash: …
- Data classes included: Owned (A), Licensed (B), Open (C) — percentages by tokens.
- Excluded classes: Ambiguous (D), Prohibited (E).
- Paywalled domains in slice: [yes/no]; if yes, licensing basis: …

- Dedup policy versions: exact v…, fuzzy v…, semantic v…
- Genre throttles applied (caps by % tokens): lyrics …, poetry …, scripts …, paywalled news …

3) Training-Time Audits

- Canary config: K=…, r ∈ {…}; registry_id: …
- Exposure results: p95=…, p99=…; max_contig_overlap=… tokens.
- Hazard-set overlaps: p95=…, p99=…
- MI testing: attacker type(s): …; AUCs by domain: news …, lyrics …, code …
- Differential privacy: (epsilon, delta)=…; accountant method: …; (C, sigma, q, steps)=…

4) Generation-Time Controls

- Decoding presets: domain-aware settings (n-gram blocks, penalties).
- Scanner stack: exact n-gram (n ranges), fuzzy (hash, thresholds), embedding model/version.
- Domain thresholds: news …, lyrics …, code … (refer to Appendix B).
- Policy routes: block/regenerate/quote rules (IDs), quote caps, paywall registry version.

5) Incident & Unlearning Readiness

- Revocation manifest support: [yes/no]; last updated: …
- Pre-taint checkpoint catalog present: [yes/no]; latest clean checkpoint: …
- Unlearning methods supported: SISA/partitioning, gradient negation, distillation; expected SLA: …
- Rollback drill date/result: …

6) Lifecycle & Governance

- Audit phases passed for this release: L0/L1/L2/L3/L4 (see §8.3); dates and report hashes: …
- WORM/Merkle log root(s) for this release window: …
- RACI sign-offs: Research (R)=…, Safety (A)=…, Legal (C)=…, Product (I)=…

7) Public Transparency (optional)

- Model card URL/id: …
- Known limitations & risky domains handling summary: …
- Rights-holder portal/opt-out path: …

Attestation

- “To the best of our knowledge and based on the attached evidence, this model was trained and is served under the controls described herein. We commit to SLAs for incident response and revocation handling as specified.”
- Authorized signatory, title, date, contact.

References

Statutes & Overviews

- 17 U.S.C. § 107 — "Limitations on exclusive rights: Fair use." LII / Cornell Law School. Legal Information Institute
- U.S. Copyright Office, *Fair Use Index* (overview and case summaries). U.S. Copyright Office

Key U.S. Cases (search/indexing vs. substitution)

- *Authors Guild v. Google, Inc.*, 804 F.3d 202 (2d Cir. 2015). (Google Books; full-text search & snippets deemed fair use.) UC Berkeley Law+1
- U.S. Copyright Office summary of *Authors Guild v. Google* (2d Cir. 2015). U.S. Copyright Office
- *Kelly v. Arriba Soft Corp.*, 336 F.3d 811 (9th Cir. 2003). (Image thumbnails as fair use.) U.S. Copyright Office+1
- *Perfect 10, Inc. v. Amazon.com, Inc.*, 508 F.3d 1146 (9th Cir. 2007). (Image search thumbnails; fair use likely.) Ninth Circuit Court of Appeals+1
- *A.V. ex rel. Vanderhye v. iParadigms, LLC*, 562 F.3d 630 (4th Cir. 2009). (Turnitin plagiarism detection; fair use.) U.S. Courts 4th Circuit+1
- *Andy Warhol Foundation for the Visual Arts, Inc. v. Goldsmith*, 143 S. Ct. 1258 (2023). (Purpose overlap narrows "transformative" analysis.) Supreme Court+1

Ongoing / Recent Litigation (illustrative)

- *The New York Times Co. v. Microsoft Corp. & OpenAI et al.*, No. 1:23-cv-11195 (S.D.N.Y.) — docket & filings. CourtListener+1
- Court's April 4, 2025 opinion on motions to dismiss in *NYT v. OpenAI/Microsoft* (S.D.N.Y.). U.S. District Court
- News coverage summarizing *NYT* allegations and posture. AP News+1

Technical Foundations (memorization, deduplication, DP, unlearning)

- Carlini, N., et al. "Extracting Training Data from Large Language Models." *USENIX Security* (2021). USENIX
- Nasr, M., et al. "Scalable Extraction of Training Data from (Production) Language Models." *ICLR* (2025) / arXiv:2311.17035. ICLR Proceedings+1

- Lee, K., et al. “Deduplicating Training Data Makes Language Models Better.” *ACL 2022* / arXiv:2107.06499. ACL Anthology+2arXiv+2
- Abadi, M., et al. “Deep Learning with Differential Privacy.” *CCS 2016* / arXiv:1607.00133. ACM Digital Library+1
- Bourtoule, L., et al. “Machine Unlearning (SISA).” arXiv:1912.03817 (2019). arXiv+1
- Carlini, N., et al. “Extracting Training Data from Diffusion Models.” *USENIX Security* (2023). USENIX

Helpful Secondary Summaries

- LII / Cornell Wex, “Fair Use” (plain-English explainer). Legal Information Institute
- Association of Research Libraries Issue Brief on the Second Circuit’s *Google Books* decision (2015). Association of Research Libraries

Note: Each citation above points to either the official opinion, a court-hosted PDF/docket, or an authoritative summary suitable for legal/policy readers.